Atari Computer Enthusists of Columbus, Ohio NOVEMBER 1985

Published by
Atari Computer Enthusiasts of Columbus, Ohio

for ACE of Columbus membership. Dues are on an annual basis and entitle the members to all club benefits (Newsletter, Disk or Tape of the month, group discounts, etc.). Monthly meetings, at DeSales High School (Cafeteria) on Karl Road are open to nonmembers.

Upcoming meeting dates at 7:30 pm are:

December 4

January 13

February 10

TREASURER
Bill Eckert
6632 Lisamarie Road
Columbus, Ohio 43229
614-891-9785

MEMBERSHIP CHAIRMAN
Tim Adcock
7544 Satterfield Road
Worthington, Ohio 43085
614-764-9492

DISK LIBRARIAN
Charles Lusco
Channing Terr. #C
Columbus, OH 43232
614-863-4016

VICE PRESIDENT
Dr Warren Lieuallen
1652 Hess Blvd.
Columbus, Ohio 43212
614-488-3977

PRESIDENT
Sheldon Wesson
444 North Pearl St.
Granville, Ohio 43023
614-587-2716

NEWSLETTER EDITOR
Mark Shuter
1854 Solera Drive
Columbus, Ohio 43229
614-899-9473

ADDRESS ALL MAIL TO
ACE of Columbus
P.O. Box 849
Worthington, Ohio 43085

Atari Computer Enthuasists of Columbus Ohio

November 11, 1985

```
***********************************
*    Special Report from the      *
*          Dr. Download           *
* Telecommunications Laboratory   *
***********************************
```

Sysops want to know:

Can machine code for the 520ST (.PRG, .RSC, .ACC files) be downloaded from CompuServe using an 8-bit Atari system, and then uploaded to a standard Atari BBS?

Can these files be downloaded from the Atari BBs using an ST, and will they execute properly?

Recent studies at the Dr. Download Telecommunications Laboratory show conclusively that the answer to these questions is YES!

Another insight gained from these experiments is that, given the current state of the ST telecommunications software, your best bet is to download the files using the 8-bit system, and send them to the ST using a cable that connects the RS232 ports of both machines. (ED.- New terminal programs have just arrived on the market for the ST, which will reduce the necessity of this procedure.)

The essentials for (relatively) foolproff ST communications are:

(1) a terminal program for the ST that supports XMODEM protocol

(2) a late-edition AMODEM feturing XMODEM compatibility with CIS, ST, and ATARI BBS XMODEM routines

(3) an RS232 interface for the 8-bit machine

(4) a cable between the RS232 ports on both machines (a null-modem connector).

We obtained CHAT 1.1, an ST terminal program, from:

SST Systems
PO Box 2315
Titusville, FL 32781
(305)269-0063 (voice)

The price was $19.95 + $2.50 shipping. This no-frills program features XMODEM, buffer capture and well organized docs.

CHAT 1.1 performs XMODEM downloads with varying degrees of relibility. It is least reliable on CIS during prime time at 1200 baud: it tends to bomb out upon encountering timing problems. We guess that htis resualts from a weak implementation of XMODEM in CHAT rather than problems with the ST.

CHAT does very well on CIS at 300 baud, and downloads properly at rither 300 or 1200 baud from the Dr. Download BBS. Other problems, such as not being able to read disk directories transmitted as strings, prevent the BBS session from being a flawless experience.

The most economical way to expedite money-intensive downloading for the ST is to do fast transfers with an 8-bit machine, using a well-developed program such as Trent Dudley's AMODEM 7.0/7.1.

This XMODEM grabs and holds on all but the worst CIS transmissions, and the error checking is much faster than CHAT's. AMODEM 7.0 is available in DL2 and on the Dr. Download BBS.

So you have your precious 32256-byte .PIC file safely depsoited on the 8-bit dump disk, and it's time to port it out to the ST to see how it looks.

Construct a cable to link the RS232 ports with the following plugs and connections:

(9-pin male)	850		ST	(25-pin female)
data out	3	<->	3	data in
data in	4		2	data out
ground	5		7	ground

Connect the RS232 ports, bring up CHAT on the ST, and toggle echo (half duplex) on.

Bring up AMODEM 7.0 on the 8-bit machine. (It is important to use this version, or another late edition, since earlier versions feature an XMODEM that blows CHAT away upon sending the last sector in the file.)

Set AMODEM for ASCII, full duplex and 1200 baud, then load the buffer. Toggle the ST for XMODEM recieve, then <START> sending. It's all downhill from here.

PICTURE.PRT

This program, written by myself, will allow printing of dot graphics. Simply RUN the program, and insert a disk with pictures on it. A menu of all .PIC extension files will be shown, and you will be asked to enter the filename ONLY (no D:,no extension). Remember the note at the bottom--when the picture is on the screen, press OPTION to print the picture, press START to return to the menu. (You may also change disks before returning to the menu, if you like.)

Many pictures need to have the colors readjusted to obtain a good print. (All solid colors wll be printed as black.) Enjoy!

SUPER FONT PLUS

This program, written by Charles Brannon, appeared on DOM 15B2, and was further enhanced by John Slaby and Warren Lieuallen. It is one of the better character set editors, in my opinion.

Its use is fairly straightforward. Upon RUNning, the main screen shows you the ATARI character set, as well as a second set which will show you your new characters. Pressing "E" will allow you to edit any character of your choice (selected via the joystick in port 0). Use the joystick to enter the character in the large grid on the screen. Pressing "E" again allows you to move on to the next character.

Many other options are available, and are fairly self-explanatory. Restore will return the character to normal; overlay will combine two characters; print will allow you to "print" some of the CHARACTERS on the bottom part of the screen by simply typing the letters on the keyboard; graphics toggles the bottom of the screen between normal text (graphics 0) and graphics modes 1 and 2. I haven't really used the antic command, so I'm not sure if it's working or not!

Just play around with SUPRFONT.PLS until you get the hang of it, and I'm sure you'll soon be designing your own fonts, too. If you do, Sheldon Wesson and I would like to get a copy of them. Thanks!

FONTMAKE.WGL

This program is very similar to SUPRFONT.PLS, but works by sending the new characters to the printer as a downloaded set, rather than to the computer's memory. Introductory information will be found on the screen when you RUN the program.

TITLE SCREEN CONSTRUCTION SET

This program appeared on DOM 21C1 as ANDER.NET. It allows characters to be drawn on the screen, and then uses a very nice rainbow color effect to create a professional-looking introductory screen. The text you want printed on the screen appears in the variable T$, which is entered in line 31. You may need to adjust the spacing to achieve the desired results.

FONTKEYS.PRT and OVERLAY

These two files are used to create keyboard overlays for the custom-designed fonts. To use them, first RUN the FONTKEYS.PRT program. This will produce a disk file called OVERLAY. Then, use the CUSTOM,PRT program to load a custom font and print the file OVERLAY using the custom font selected.

The result will be a representation of the Atari keyboard, showing all the custom-designed characters (both upper and lower case) within the "keys". However, if some of the Atari graphics characters have been redefined (for example, the control-Q character), the "boxes" around the characters may not appear correct. At present, the only way to fix this is to change the graphic characters back into their normal patterns, or just live with the funny-looking boxes!

Warren Lieuallen

A Piece of the ACTION!

By Dave Beck

Variables

Last month's article gave a short introduction to the ACTION! programming language, showing some of its advantages and disadvantages. This month's article is the first of several going into the details of the language. For those of you who are considering another language to use as an alternative to Assembler or BASIC, these articles should give you a solid base of understanding to determine if this is the language for you. In light of this, and in order to allow a more solid understanding of how this language works, I will be doing comparisons to Assembler and BASIC to a large degree.

The basic variable types in ACTION! are BYTE, INT and CARD. In addition to these basic variable types, there are two extensions to them: all three types can be defined as arrays, and pointers can be defined to 'point' at them. Besides the flexibility these types and extensions give you, there is also the ability to create your own variable types from combinations and variations of the basic ones.

- BYTE - This is a one byte (eight bit) area which can contain the value 0-255, ACTION! considers a BYTE variable to always be positive. For those familiar with C or Pascal, this type can also be called CHAR. Although the value of a BYTE/CHAR variable is numeric, it can be compared/assigned to an ATASCII character by preceding the character with a single quote mark. The code below demonstrates this.

```
BYTE Test_Char

     Test_Char = 'A
     IF Test_Char='A THEN
       Test_Char='A+64
       ...
```

  This example also demonstrates the format for a variable name, it must begin with an alphabetic character and may contain upper and lower case alpha, numeric and the underscore character.

- CARD - This is a two-byte variable which always contains a positive value. It can range from 0 to 65535 and is stored in LSB/MSB format. As this is the same format that the 6502 processor uses to store addresses, this variable type can be handy for handling addresses and vectors used by the operating system.

- INT - This is a two-byte variable which can contain 32767 to -32768. It is stored in LSB/MSB format.

ACTION! does no checking for overflow of any of the basic types in numeric use as BASIC does. Near the boundaries of each type strange things can happen: adding 1 to a BYTE of 255, a Card of 65535 and an INT of 32767 will result in a BYTE of 0, a CARD of 0 and an INT of -32768. This means that all boundary checking must be done in the program.

When it comes to defining variables, ACTION! is much more like Assembler than BASIC. All ACTION! variables must be defined before they can be used, unlike BASIC in which numeric variables are defined implicitly the first time they are used in the program. While this makes a little more work for the programmer, it also helps reduce program errors due to misspelling. For example:

```
BASIC

10 C0UNT=1536
20 POKE COUNT,0
30 COUNT=COUNT+1
40 IF COUNT<1541 THEN 20
50 END
```

```
ACTION!

BYTE COUNT
PROC ADDUP()
C0UNT=1536
DO
WHILE COUNT<1541
  Poke(COUNT,0)
  COUNT==+1
OD
RETURN
```

In both cases, the intent was to put 0 in memory locations 1536 through 1540, but in the assignment of 1536 to the variable the variable name is misspelled with a 0 instead of an O, a common error. When the ACTION! program is compiled, the error will be indicated because COUNT has not been defined. The BASIC program will not catch the error, and if run will put 0 in the first 1540 locations of memory. This will cause the computer to lock up and even if you have saved the program so you don′t lose it when you have to re-boot, it may take a while to find the misspelling. I realize that there are more efficient and less error-prone ways to accomplish the above task in both languages, but the point is still valid.

Also like assembler are the three different ways that a variable can be defined: simply defining it to exist without concern as to where is it exists in memory or what is contains initially (the way all BASIC variables are set up), defining it to be located at a specific place in memory, and defining it to be initialized with a specific value (all BASIC numeric variables are 0 to start and strings are spaces, no choice). Examples and an explanation of the advantages of each are as follows:

```
BYTE ARRAY counter(5),
           pmcolor_shadow=704,
           pmcolor_direct=$D012,
           text_string="Init",
           time_limit=[20, 15, 10, 5]
```

The first variable, counter, will be set up somewhere in memory, but we really don't care where. The second two variables reference the player-missle color registers and there are several points that need to be made about them. First, notice that one address is set in decimal and the other is set in hex; ACTION! allows you either form so you can use whichever you are most comfortable with. Also you will notice that there is no length assigned to these arrays, these are simply the start point, you need to keep track of the length yourself. The last two variables are examples of initialization at compile time. The advantages here are 1) the initial value can be easily seen as it is in the same statement as the variable definition rather than later in the program, and 2) only a minimum amount of storage is needed, in BASIC, space would be reserved for the variable and more space would be required for the code to move the value into it.

The array named text_string is a bit different from the others by virtue of its initialization. When the compiler sees "s around the value, it sets the array up as a special type known as a string. To illustrate the difference, I'll compare the way text_string and time_limit are set up in memory: time limit is a 4 byte array containing the values 20, 15, 10 and 5, no surprise there; but text_string is a 5 byte array. The last 4 bytes contain the ATASCII values for the characters I, n, i and t, the first byte contains the length of the string, in this case a 5. This allows the use of ACTION! subroutines to handle the entire string rather than having to do things like printing, one byte at a time. As a result of this length convention, a BYTE ARRAY can be any length up to the limit of memory, but a string can only be 255 bytes long.

POINTER variables are a whole new animal to BASIC programmers, but assembler programmers may see their similarity to the use of index registers. A pointer is exactly what it says it is, a pointer to an address in memory. There are three types, BYTE POINTER, CARD POINTER and INT POINTER. The pointer can be assigned to point to a particular place in memory directly or to an existing variable ie:

```
BYTE pm1color=704
BYTE POINTER pntr

pntr=704          ;these two lines do *exactly*
pntr=@pm1color    ;the same thing.

pmcolor=47        ;so do
pntr^=47          ;these
```

The @ in @pm1color means 'address of', much as the ADR function in BASIC returns the address of a variable, so what we're doing here is pointing the BYTE POINTER pntr to the variable pm1color. The pntr^ used later means 'use the BYTE (remember, pntr is a BYTE pointer) that pntr is pointed to', not pntr itself.

As you can see, variables in ACTION! can be more difficult to use than they are in BASIC, but with that increased difficulty comes a significant degree of power. Once the techniques are learned, things that are nearly impossible in basic become almost simple to do in ACTION!. I've left out some of the more exotic features of arrays and the ability to define your own record types from this article, we'll cover these things later, after I learn a little more about them myself. Next month I'll discuss the logic structures of the language that help make this one of the more easily understood programming languages for the ATARI.

PHYSICS FOR FIVE YEAR OLDS
by Senator Morris

I wish to submit a challenge to the other members.

Since that Atari Computer Enthusiast of Columbus, Ohio is a family club, it is not unreasonable to assume that at least one member of your household is literate. If we follow that logic a bit farther, it is possible someone living at your house can read and write.

Face it. It is our responsibility as Enthusiast to educate all the cyberphobes. So, let's all get together and share some of those early experiences through which we went before we arrived where we now are; and, at the same time, we might also learn something that will help us get where we want to go.

We are standing on the threshold of a renaisance. Legends are dying and records are being broken. Never in the history of civilization has so much taken place in such a brief period of time.

Our new President Sheldon is telling the truth. Up to now, we were merely concerned with what is available. Now we are headed in a new direction. When we say "user friendly," we mean "USER FRIENDLY."

Take just a moment and consider the year 1985. It isn't even over yet and there has been famine, drought, earthquakes, tornades, hurricanes, plane disasters, hostage taking; I mean it, they say the kids today are different and you can see why. You name it, it has happened this year. Heck, you can't even trust the spies. Look at how many moles they discovered this year. I'm talking about serious turmoil.

Maybe you can offer some of the exercises that help you to cope or tell how you deal with the inconsistency of day-to-day boredom. What is the best news you've heard this year? What about Wilma Mankiller? Bless her heart. Even the Indians (Native Americans, not Asian) are permitting a woman to become their chief this year.

When someone asks me to describe myself, I say that I am a "feminist journalist." I'll tell you why. Most of the children I know either live with their mother or father and few actually see each parent every day. Myself, I spend a greater amount of time alone with my mother than most kids spend with their mother and father together. In addition to the time I spend with my mother, I spend more time than that with my grandmother. The kicker is that I spend more time with my father than I spend with my mother and my grandmother together. Confusing? Now you know why I am like I am.

I don't want you to feel sorry for me. Just wanted to let you know a bit about me. My life has been pretty much the same since birth. My mother wanted to experience "home birth" and wasn't successful, so after several hours, she decided to go to the hospital. After birth, I was confiscated by the hospital and immediately placed under the care of a physician who prescribed medical attention that included incarceration for ten days while antibiotics were administered intravenously. All the while this was happening to me, my parents were charged with child neglect.

The most absurd aspect of my "royal birth" was the fact that my father was prohibited from experiencing the bonding experience. Then, he was insulted in writing with the accusation by the hospital with being "unknown and without involvement," both. All this happened after he gave the hospital a check for $400 and a promise to pay the remainder of the charges.

Lately, my life is much of the same. I love my Atari 400, but should I receive a Commodore Amiga for Christmas (or, for that matter, Hanukkah), it wouldn't make me sad.

What is your latest interest? Mine would probably be "mood-setting." The current buzz word I like is "pscho-synthetic persuasion." Look around you. All your friends have the state of the arts equipment and want something better. Imagine having an analogue reproduction with a sound that can possibly create mental images and sight pictures with each seaparate sound. Have you seen a compact synthesizer? This has to be the greatest time in the history of the human race to live in. And we can contribute. It is easy. By submitting articles to "OUR NEWSLETTER."

My sermon is over. My article begins.

Back to the subject. What kind of a year has it been for me? Needless to say, this is the Chinese Year of the Bull. (Somehow I believe that the name of the year has been shortened! Right? Now, everyone together: Right? Right!)

Anyway, in January I attended the only presidential inaugaration ever cancelled. In March I went to see the New York St. Patrick's Parade only to discover that it has been held a day early. Nothing much happened till July. At that time my father let me sleep through all the parades except the Doodah Parade. And to atone for his indiscretion, we were the only ones skating in it using canes. Talk about embarrassing. The guy with the tv camera can always find my father and me.

My dad is the president of an organization called the "Homely Dandelions." I tell him it is just oversensitivity from being a shy person. You've got to pretend more . . . if you act like a toad, people will know you are a toad.

AFTER AND BEFORE

The procedure we follow as scientists begins with the initial observation. (I am a member of the New York and Ohio Junior Academy of Science and will share publications with anyone who is a member of AAAS.) As a scientist, I begin with attention to detail. We look first to see how it behaves. We hunt for various examples and gather as many facts about our project as we can and then we begin to classify the results of these accurate observations.. From each of these classifications, we draw certain general conclusions which we can call laws or rules.

My curriculum of study is limited to and includes physics, astonomy, chemistry, mathematics, geography, history and, this year, grammar. The grammarian is still another type of scientist. You might say that grammar is a descriptive statement explaining the way a language works.

I'll close with a casual observation: We have the best Atari Club, so, let's put out a newsletter that all other clubs will want to imitate.

LESSON #1
What is a penguin?
It is a bird which doesn't fly.
What is an auk or guillemot??
Diving birds.

Remember: "Life is a multi-layered tort" and when an idea, a thought, a concept, needs a symbol for expression, the difficulty arises as to what specific term, understandable to others, ought to be used. What is the problem -- communication.

So, in all you will do, the best you can do is the least you should do. And?

If more than 100% you could do, then that is okay too.

It has been a real pleasure and a great honor to be a member of ACE of Columbus and all of us should contribute whatever we can to show Sheldon that we care.